



Optimal real-time resonant scanner linearization using filtered Hermite interpolation

MICHAEL G. GIACOMELLI* 

Department of Biomedical Engineering, University of Rochester, 207 Goergen Hall, Box 270168, Rochester, NY 14627, USA

**mgiacome@ur.rochester.edu*

Abstract: High-speed laser scanning microscopy frequently relies on resonant scanners due to their order of magnitude increase in imaging rate compared to conventional galvanometer scanners. However, the use of a nonlinear scan trajectory introduces distortion that must be corrected. This manuscript derives a new algorithm based on filtered Hermite polynomial interpolation that provides the optimal shot-noise-limited SNR for a fixed number of photons and provides higher spatial accuracy than previous methods. An open-source library is presented using the Intel advanced vector instruction set (AVX) to process up to 32 samples in parallel. Using this approach, I simultaneously demonstrate lower shot noise variance, moderately higher spatial accuracy and greater than 1 gigapixel per second interpolation rate on a desktop CPU.

© 2023 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

Resonant scanning mirrors combine very high-speed scanning over a wide angular range with excellent light throughput. As result, they have become widely used in laser scanning techniques such as confocal microscopy, multiphoton microscopy, STED microscopy, LIDAR, among many others. However, resonant scanners obtain their high speed and angular deflection by oscillating through a sinusoidal trajectory. As a result, they are bidirectional scanners that produce a warped image in which pixels have a time-varying width. In contrast to polygonal scanners and many galvanometers, which are unidirectional and produce a more or less undistorted image, at a minimum each odd scan line must be reversed to produce an interpretable image. Additional steps may also be required to compensate for the nonlinear (sinusoidal) scanner trajectory.

The literature on the correction of resonant scanner images dates back many decades [1]. The simplest solution is to limit imaging to the central part of the resonant scan where the trajectory is somewhat more linear [2]. However, this approach is rarely used because it wastes much of the scan cycle and still results in a modestly distorted image that is typically undesirable for microscopic imaging. Historically, another extremely common approach was to use a frame grabber that could accept a nonuniform pixel clock. Analog circuitry was then used to generate pixels at time intervals that canceled out the variation in the resonant scanner velocity, resulting in a uniformly sampled image [3]. The advantage of this approach is that the only required computation is the reversal of the backwards half of the scan, and so it could be used with early computers. However, as imaging speeds have increased, frame grabbers have been replaced with conventional A/D converters that are usually not compatible with highly nonuniform sampling clocks. There are more subtle disadvantages of the analog pixel clock as well. First, the hardware pixel clock is typically synchronized to a uniform system clock inside the device, thus the continuously distributed true pixel locations are rounded to the nearest neighboring clock cycle, resulting in distortion of the image point spread function [1]. Second, during the oversampled portion on the edges of the scan, all photons not lining up with the pixel clock samples are discarded, resulting in loss of sensitivity. Thus, this approach is clearly not optimal.

An alternative approach favored in more recent publications is to perform numerical correction to resample the nonuniform resonant points into a uniformly sampled line. While it is straightforward to derive an relationship between samples along the resonant scan and pixels in the desired rectilinear image [4], interpolation is always required because the uniformly spaced pixels will not correspond to integer samples in the resonant scan. One approach is to simply round fractional pixel values to the nearest neighboring sample [2,5]. While this has the approach of low computational complexity, it retains the disadvantage of rounding to the nearest sample in analog pixel circuits. Indeed, analysis of nearest neighbor interpolation shows that it introduces substantial loss of high frequencies and reduced SNR relative to higher order interpolation [6]. Linear interpolation is also widely used [4], and has the advantage of much higher accuracy for only a minor increase in complexity. Less obvious, linear interpolation incorporates information from up to two points, and thus provides higher noise rejection because shot noise between the two points is partially averaged. However, while linear interpolation is superior to nearest neighbor interpolation, it becomes less accurate further from integer index values. Finally, all forms of interpolation have limited order, and thus can only incorporate photons from a limited number of samples, resulting in wasted photons at the edge of images.

A general disadvantage of numerical interpolation is the relatively high computational cost. Sampling rates in resonant scanning are on the order of 80-140 MHz [7,8] and typically interpolate 1024 or 2048 pixels per direction, per line. For a 12 KHz (unidirectional) scan rate, this equates to 25-50 Mpixel/s. This cost is then compounded across all channels, of which there are typically at least 2-4. Thus, a system may be expected to interpolate 100s of Mpixel per second in real-time. One approach to this has been to show only the distorted resonant samples with no correction [2,9] or to use lower order interpolation [5] for real-time preview and then reconstruct higher resolution images only in post-processing. While more computationally efficient, this approach is complicated, results in users seeing a distorted image during acquisition, and saves significant redundant information to disk, bloating file sizes. Conversely, some commercial products have adopted FPGA acceleration, adding dedicated hardware to perform interpolation. However, this approach is complex, while restricting flexibility as the fixed FPGA hardware will typically only be capable of processing certain numbers of channels and frame sizes.

An alternative approach to accelerating processing is to use the single instruction, multiple data (SIMD) hardware units built into all modern CPUs. This hardware can perform common operations such as multiplication on batches of samples in parallel, vastly accelerating processing of tasks such as filtering and interpolation that are composed of repetitive operations on data. The Intel AVX instructions operate on 256 bit vectors which can contain both floating point (AVX1) and integer (AVX2) data types as well as 512 bit vectors (AVX512) [10]. For 16 bit pixel values, this corresponds to 16 and 32 elements per vector, respectively. Previous work has demonstrated order of magnitude increases in throughput processing image data [11], suggesting that radical improvements in processing might be possible for resonant scanning systems.

In this work, I first derive an expression for the mapping rectilinear pixel coordinate to resonant sample number. Using this expression, I calculate the Nyquist sampling rate as a function of pixel number, and then compute per-pixel finite impulse response (FIR) filter coefficients that will optimally filter samples to incorporate all available photon information. Next, I implement 4-tap cubic Hermite polynomial interpolation, which enables greater accuracy, less loss of resolution and higher SNR. Finally, I provide an open-source implementation using SIMD processing, demonstrating more than 1 Gigavoxel/s interpolation rate on a single CPU core. I show that this algorithm and implementation enable real-time processing with higher SNR and accuracy than previous approaches.

2. Derivation

The derivation of Haji-Saeed et al. [4] provides an excellent overview of the forward transformation from sinusoidal trajectory to linear. I proceed by inverting this transformation, developing an expression for the raw (sinusoidal) sample values that can be fit to a polynomial to yield the linearly spaced pixels. As explained in [4], a bidirectional linear scan trajectory with n uniformly spaced pixels over focal length r , angle θ_0 for the L' pixel has the form:

$$y(L') = \begin{cases} -r\theta_0 + \frac{2r\theta_0 L'}{n}, & (L' < n) \\ r\theta_0 - \frac{2r\theta_0(L'-n)}{n}, & (L' \geq n) \end{cases} \quad (1)$$

where the first n pixels represent the forward scan, and the next n represent the backward scan returning to the initial position. Conversely, a resonant scan over m nonuniform forward and m backward *samples* for the L sample has the form:

$$y(L) = -r\theta_0 \cos \frac{2\pi L}{m}, \quad (0 \leq L' \leq 2m) \quad (2)$$

Equating these expressions for position y :

$$\begin{cases} -r\theta_0 + \frac{2r\theta_0 L'}{n} = -r\theta_0 \cos \frac{2\pi L}{m}, & (L' < n) \\ r\theta_0 - \frac{2r\theta_0(L'-n)}{n} = -r\theta_0 \cos \frac{2\pi L}{m}, & (L' \geq n) \end{cases} \quad (3)$$

Rearranging terms, taking the arccosine, and then solving for L as a function of L' :

$$L = \begin{cases} \frac{m}{2\pi} \cos^{-1} \left[1 - \frac{2L'}{n} \right], & (L' < n) \\ \frac{m}{2} + \frac{m}{2\pi} \cos^{-1} \left[\frac{2}{n}(L' - n) - 1 \right], & (L' \geq n) \end{cases} \quad (4)$$

This expression relates the resonant (L) sample indices to the linear (L') pixel numbers. Thus, given a pixel number, the corresponding (generally non-integer) sample index can be easily calculated (Fig. 1(A)). Furthermore, since L is a function of both m (the number of samples recorded for each forward and backward resonant sweep) and n (the number of linearly spaced pixels to reconstruct), this equation is fully general for any sampling rate, resonant scanner frequency or number of pixels to be calculated per frame. An additional feature of this derivation is that it enables calculating the degree to which each pixel in the output image is oversampled which can be used to reject excess shot noise during linearization by filtering photon counts from multiple samples before interpolation. Figure 1(B) plots the change in the input sample number for each output pixel, showing the large change in oversampling over the course of each scan. Areas containing spatial frequencies in green should be retained while those in red contain only broadband shot noise and should ideally be rejected during linearization.

The sample indices calculated with Eq. (4) are in general not integers and thus interpolation is required to calculate pixel values at non-integer samples. Furthermore, out-of-band shot noise is present in each sample that should be removed prior to linearization in order to maximize sensitivity. This is implemented with a two-step approach. First, for each input sample, the maximum spatial frequency (MSF) relative to the center of the field of view is calculated as the absolute inverse difference between the L and $L + 1$ samples:

$$MSF = \left| \frac{1}{y(L) - y(L-1)} \right| \quad (5)$$

Second, for each MSF, an odd-length $N + 1$ tap FIR filter is specified with a cut off frequency f_{3dB} according to:

$$N = \text{RoundUpToEvenNumber}(3/MSF) \quad (6)$$

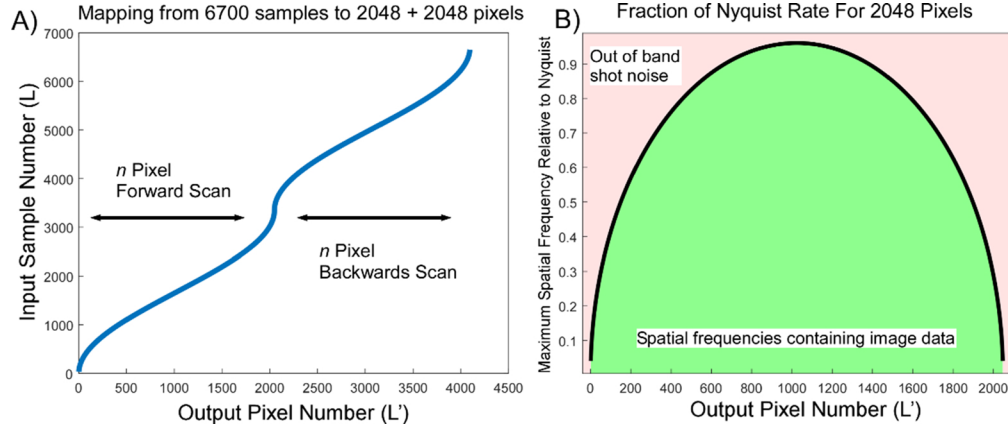


Fig. 1. A) Relationship between input sample number and output pixel number. B) Change in the output pixel number relative to the input sample number. As the scanner velocity slows towards the edges while the sampling rate remains constant, the maximum spatial frequency in the image diminishes relative to the Nyquist sampling rate while the shot noise power spectrum remains uniform. As a result, pixels towards the edges contain out of band shot noise (red area) that can be removed during interpolation without loss of signal (green area).

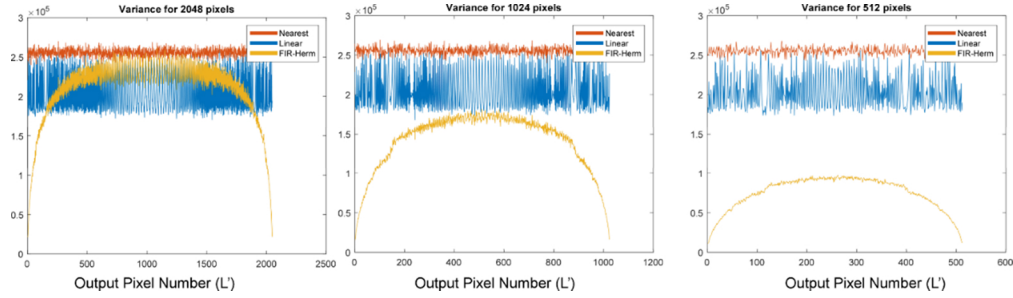


Fig. 2. Variance calculated over 4 frames for interpolation to 2048, 1024 and 512 pixels using nearest neighbor, linear and filtered Hermite interpolation. Filtered Hermite shows dramatically lower variance at the edges and for lower pixel counts than the other methods. Nearest neighbor is simply the raw shot noise variance. Linear interpolation oscillates between the same variance as nearest neighbor (integer indices which reduce to nearest neighbor) and a lower value (half-integer indices which average two adjacent samples but attenuate higher frequencies).

$$f_{3dB} = MSF + \frac{1.75}{N + 1} \quad (7)$$

where $1.75/N$ represents the approximate half width of the transition band of a Hamming windowed low pass filter and odd length is used to enable zero-delay filtering. However, for pixels with spatial frequencies close to unity, N is small, which results in a transition band edge at greater than Nyquist. Thus, if $1.75/N$ is greater than 0.25, N is iteratively increased until it falls below that threshold. Next, the windowed-sinc method with a Hamming window is used to calculate the FIR coefficients and then filter the sample data [12]:

$$FIR_{Coefs} = WindowedSincCoefficients(f_{3dB}, N + 1) \quad (8)$$

$$y_{filtered}(L') = Filter\left(FIR_{Coefs}, y\left(L' - \frac{N}{2} \dots L' + \frac{N}{2}\right)\right) \quad (9)$$

where $\text{FIR}_{\text{Coeffs}}$ are the $N + 1$ FIR filter tap coefficients and L_{filtered} is the low pass filtered sample with out of band shot noise removed. Notably, because the filter taps are odd, the $L' - \frac{N}{2} \dots L' + \frac{N}{2}$ sample indices result in a filter with no delay relative to L' . Finally, cubic Hermite interpolation over the filtered points enables calculation of the value of y_{filtered} at non-integer values L :

$$Y_{\text{interpolated}}(L) = \text{CubicHermiteInterp}(y_{\text{filtered}}(L' - 3 \dots L', L)) \quad (10)$$

CubicHermiteInterp:

$$\begin{aligned} \text{frac} &= L' - \text{floor}(L'); \\ Y &= (3 * (y(L' - 2) - y(L' - 1)) + y(L') - y(L' - 3)) / 2; \\ Y &= (Y * \text{frac}); \\ Y &= Y + 2 * y(L' - 1) + y(L' - 3) - ((5 * y(L' - 2) + y(L')) / 2); \\ Y &= (Y * \text{frac}); \\ Y &= Y + (y(L' - 1) - y(L' - 3)) / 2; \\ Y &= (Y * \text{frac}); \\ Y &= Y + y(L' - 2); \end{aligned} \quad (11)$$

Consequently, an entire resonant scan can be linearized by performing this operation n times to compute the forward direction, skipping ahead n pixels, and then stepping backwards (due to the reverse order of the backwards scan) performing the operation a further n times from pixel $2n-1$ to n .

3. Open-source implementation

At 2048 pixels per line, the above algorithm requires $\sim 88,800$ FIR taps per resonant scan per spectral channel and a further 14 floating point operations per interpolated pixel per spectral channel, or about 9 billion floating point operations per second (FLOPs) when operated in 4-channel mode with a 12 KHz resonant scanner. This further increases to 36 billion FLOPs when operated in 16-channel mode. I developed an open-source library that implements the above algorithm using AVX intrinsics, operating at > 1 gigapixel/s, it can perform 16-channel interpolation at faster than 1.5 times real-time on a single core of an Intel i7-12700k with AVX512 and FLOAT16 hardware extension enabled.

The implementation precomputes all FIR taps and all L indices and assumes that 4-channel or 16-channel sample data is provided as interleaved 16-bit samples as is typical for nearly all high-speed digitizers. In 16-channel mode, all 16 channels from 2 samples are loaded into a single 512-bit ZMM register, are normalized and then converted into half precision (float16) format. A pair of FIR tap coefficients is loaded and broadcast into 16 positions each of a ZMM register. Finally, a single fused-multiply operation calculates all 32 taps in one cycle. In 4-channel mode, 4 samples are processed in parallel using a 256-bit YMM register. As a result of the narrower register, throughput is lower in 4-channel mode, but overall frame rate is higher due to less data per second.

After filtering each sample, the algorithm checks to see if all 4 filtered samples required for the next interpolation are available with the goal of interpolating immediately while the filtered data is still in local cache to avoid memory latency. In 16-channel mode, all 16 channels from the 4 samples needed for 4-tap cubic Hermite are loaded into 4 YMM registers. Interpolation then proceeds on 16 element YMM vectors. This repeats 16 times, yielding 16 samples that are 16-way interleaved. These are loaded into ZMM registers and permuted to deinterleave before

finally scattering to 16 separate channel buffers 256 bits at a time. In testing, the larger write size was essential to efficiently utilizing memory bandwidth. In 4-channel mode, processing is more complex because 4 sequential samples of 4 channels each must be permuted 4 times each to obtain the 4 packed YMM registers. Interpolation then proceeds as in the 16-channel case, except with 4 channel buffers written 256 bits at a time. In either case, upon reaching the end of the forward sweep, the software moves the output pointer to the end of the line of pixels and then fills in backwards sweep from the right-most pixel moving backwards in memory towards the left-most pixel in order to account for the reversed direction on the backwards sweep.

Example raw image files and source code in both MATLAB and AVX are available at [23].

4. Results

4.1. Performance

Table 1 summarizes the performance of the library. In testing, both configurations were real-time, with 16-channel being more optimized and avoiding the need for some permute operations due to larger continuous data size, but 4-channel having faster frame times due to less data. I further confirmed good scaling out to 4 cores in 16-channel mode (> 4 billion pixels/s), beyond which the bandwidth limitations of DDR4 limited scaling. In real-time operation on a microscope in which the digitizer and other software compete for RAM bandwidth, I found the fastest processing with 3 parallel threads.

Table 1. Linearization performance on a Core i7-12700 k with DDR4-3000 RAM.

Channels	Millions of pixels per second	Frame time (2048 × 2048)
4	546	29 ms
16	1154	54 ms

4.2. Sensitivity

I next characterize the sensitivity improvement from filtered Hermite interpolation by recording uniform fluorescent targets using a two photon microscope described previously [7]. In brief, a 12 KHz galvo-resonant scan head (LSK-GR12, Thorlabs) using the Novanta CRS-12 k scanner and a FemtoFiber ultra 920 laser excited a fluorescent target. Fluorescence was detected with using high dynamic range silicon photomultipliers [13] and then digitized with an AlazarTech ATS9416 in 16-channel mode. Following recording, the raw sample data was processed and the shot noise variance of the uniform target calculated. As expected, filtered Hermite interpolation provides significantly lower shot noise variance with oversampled data.

4.3. Accuracy

Next fluorescent beads were imaged, and data sets linearized with both filtered Hermite and linear interpolation (Fig. 3). As anticipated, while the overall accuracy of linear interpolation was high, and it provided nearly identical results to higher order Hermite interpolation for indices close to integers (green arrows), it deviated otherwise (red arrows), resulting in distorted signal amplitude and point spread function.

4.4. Biological imaging

Finally, I imaged human skin excisions obtained under a protocol approved by the University of Rochester Medical Center Research Subjects Review Board. Figure 4 shows the raw sample data, the deinterleaved and linearized data as well as cropped areas processed with filtered Hermite and linear interpolation.

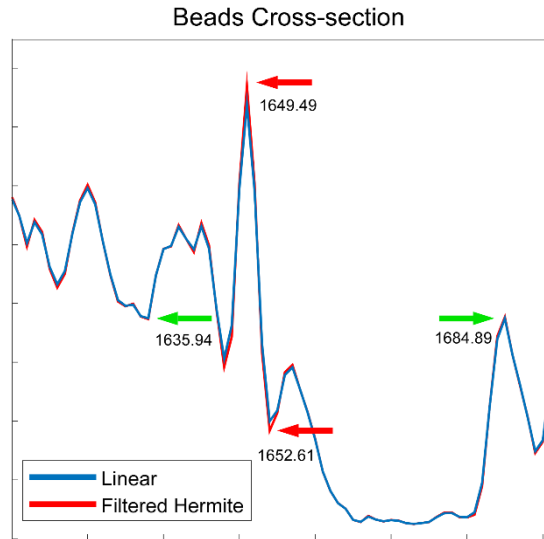


Fig. 3. Fluorescent beads linearized with linear and filtered Hermite interpolation at the center of a 2048 pixel field of view. Linear interpolation frequency response varies randomly based on the fraction part of the index. Some peaks (green) correspond to near-integer sample indices and closely follow filtered Hermite. Others (red) have near-half-integer sample indices and attenuate the extreme positions through averaging.

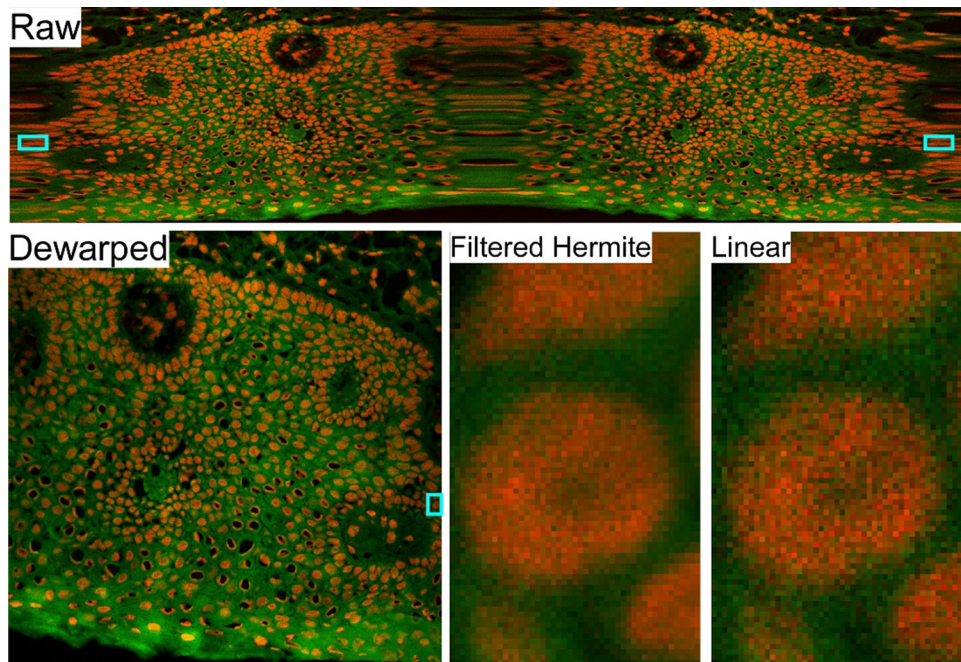


Fig. 4. 1024 bidirectional lines of human skin. Raw shows the 1024 bidirectional scan lines of 6656 samples (3328 per direction) returned by the ADC. The dewarped frame shows the linearized data processed with filtered Hermite interpolation to produce 2048×2048 pixels. Filtered Hermite and linear insets show the blue box processed with both filtered Hermite and linear interpolation, showing reduced shot noise at the image edges due to the filtered Hermite algorithm.

5. Discussion

Linearization of resonant scanner images is an important but often overlooked aspect of high-speed imaging in applications such as neuroscience [14–16], laser ophthalmoscopy [17,18], and super-resolution microscopy [19–22] microscopy. The lack of a high-accuracy and high-speed reference implementation has resulted in a variety of *ad hoc* solutions that have limited performance and/or accuracy. To address this, I build on previous work to derive an algorithm that is optimal from the point of view of extracting maximum shot noise limited sensitivity while retaining high accuracy and then provide a highly optimized C reference implementation as well as MATLAB example code. While I focused on implementation on x86 PCs, the algorithm uses only a small amount of memory and could be readily implemented on an FPGA.

In practical terms, the above algorithm has some advantages over the nearest neighbor and linear interpolation proposed previously. First, sensitivity is dramatically improved at the edge of images, where SNR is typically lowest in applications such as two photon microscopy and confocal microscopy due to field-dependent optical aberrations. Improving SNR at the edges can increase maximum usable field of view and facilitate more accurate stitching of mosaic images by providing higher SNR for overlapped pixels [7]. Second, if reconstructed images are undersampled or if a system is operating with a high sampling clock (for example, to minimize sample jitter), the SNR improvement is extended to even the center of the field of view, improving image sensitivity and/or maximum imaging depth. Finally, the poor (nearest neighbor) or inconsistent (linear) spatial frequency reproduction of common methods is avoided, which may be important in applications such as deconvolution, adaptive optics and quantitative imaging where a pixel-dependent point spread function is undesirable. The disadvantage of this method is that more multiplications are required per pixel, but this is overcome using SIMD processing.

It is interesting to note that SNR and point spread function for common interpolation methods varies from pixel to pixel based on how close each pixel is to a whole sample index as demonstrated in Fig. 2 and Fig. 3. In contrast, this algorithm provides a more uniform point spread function and an SNR that varies gradually across the image field and to an extent that can be configured for the application. While the goal of this work was optimal shot-noise-limited SNR for a fixed number of photons, if a more uniform SNR were required (e.g. for deconvolution), f_{3dB} in Eq. (7) could be constrained or set to a constant, resulting in a more or even completely uniform SNR at the expense of some sensitivity.

6. Conclusion

I derive and then implement in an open-source library an algorithm for resonant scanner linearization that is optimal from the point of view of maximizing sensitivity while providing very high accuracy using higher order interpolation. SIMD instructions are used to achieve faster than real-time processing. Adoption of this algorithm would improve sensitivity and accuracy of high speed confocal and two photon imaging in many applications.

This study was supported by the U.S. National Institutes of Health, Grant No. R37-CA258376 and R21-EB032839.

Funding. National Institute of Biomedical Imaging and Bioengineering (R21-EB032839); National Cancer Institute (R37-CA258376).

Acknowledgments. I thank George Funkenbusch for his initial help with AVX programming and Vincent Ching-Roa for providing the skin raw image data.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available in Ref. [23].

References

1. D. G. Tweed, "Resonant Scanner Linearization Techniques," *Laser Scanning Rec.* **0498**, 161 (1984).

2. M. J. Sanderson and I. Parker, "Video-rate confocal microscopy," *Methods Enzymol.* **360**, 447–481 (2003).
3. J. Xiang, Z. Wu, P. Zhang, D. Huang, and G. Wei, "The precision improvement of the scanner in optical scanning imaging system," *Opt. Laser Technol.* **30**(2), 109–112 (1998).
4. B. Haji-Saeed, J. Khoury, C. L. Woods, D. Pyburn, S. K. Sengupta, and J. Kierstead, "Mapping approach for image correction and processing for bidirectional resonant scanners," *Opt. Eng.* **46**(2), 027007 (2007).
5. A. Cheng, J. T. Gonçalves, P. Golshani, K. Arisaka, and C. Portera-Cailliau, "Simultaneous two-photon calcium imaging at different depths with spatiotemporal multiplexing," *Nat. Methods* **8**(2), 139–142 (2011).
6. D. K. Wise and R. Bristow-Johnson, "Performance of low-order polynomial interpolators in the presence of oversampled input," *Proc. 107th AES Conv.* (1999).
7. C. Huang, V. Ching-Roa, Y. Liu, and M. G. Giacomelli, "High-speed mosaic imaging using scanner-synchronized stage position sampling," *J. Biomed. Opt.* **27**(01), 1–12 (2022).
8. J. Thomas, D. Megherbi, P. Sliney, D. Pyburn, S. Sengupta, J. Khoury, C. Woods, and J. Kierstead, "An FPGA-based method for a reconfigurable and compact scanner controller," in *Proc. SPIE 5873, Optical Scanning 2005* (2005), Vol. 5873, p. 121.
9. D. Langer, M. van 't Hoff, A. J. Keller, C. Nagaraja, O. A. Pfäffli, M. Göldi, H. Kasper, and F. Helmchen, "HelioScan, A software framework for controlling in vivo microscopy setups with high hardware flexibility, functional diversity and extendibility," *J. Neurosci. Methods* **215**(1), 38–52 (2013).
10. C. Lomont, "Introduction to Intel Advanced Vector Extensions," Intel White Paper 23 (2011), pp. 1–21.
11. M. Moradifar and A. Shahbahrani, "Performance Improvement of Gaussian Filter using SIMD Technology," *B.-P. High-Pass Iran. Conf. Mach. Vis. Image Process. MVIP 2020-Febru.* (2020).
12. S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," *Digit. Signal Process.* **1**, 285–296 (2013).
13. V. D. Ching-Roa, E. M. Olson, S. F. Ibrahim, R. Torres, and M. G. Giacomelli, "Ultrahigh-speed point scanning two-photon microscopy using high dynamic range silicon photomultipliers," *Sci. Rep.* **11**(1), 5248 (2021).
14. N. J. Sofroniew, D. Flickinger, J. King, and K. Svoboda, "A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging," *eLife* **5**, 1–20 (2016).
15. C. Tischbirek, A. Birkner, H. Jia, B. Sakmann, and A. Konnerth, "Deep two-photon brain imaging with a red-shifted fluorometric Ca²⁺ indicator," *Proc. Natl. Acad. Sci. U. S. A.* **112**(36), 11377–11382 (2015).
16. B. Li, C. Wu, M. Wang, K. Charan, and C. Xu, "An adaptive excitation source for high-speed multiphoton microscopy," *Nat. Methods* **17**(2), 163–166 (2020).
17. A. Dubra and Y. Sulai, "Reflective afocal broadband adaptive optics scanning ophthalmoscope," *Biomed. Opt. Express* **2**(6), 1757 (2011).
18. Q. Yang, L. Yin, K. Nozato, J. Zhang, K. Saito, W. H. Merigan, D. R. Williams, and E. A. Rossi, "Calibration-free sinusoidal rectification and uniform retinal irradiance in scanning light ophthalmoscopy," *Opt. Lett.* **40**(1), 85 (2015).
19. P. F. G. Rodríguez, Y. Wu, H. Singh, and H. Zhao, "Building a fast scanning stimulated emission depletion microscope: a step by step guide," *Curr. Microsc. Contrib. to Adv. Sci. Technol.* **2**, 791–800 (2012).
20. X. Wu, L. Toro, E. Stefani, and Y. Wu, "Ultrafast photon counting applied to resonant scanning STED microscopy," *J. Microsc.* **257**(1), 31–38 (2015).
21. L. K. Schroeder, A. E. S. Barentine, H. Merta, S. Schweighofer, Y. Zhang, D. Baddeley, J. Bewersdorf, and S. Bahmanyar, "Dynamic nanoscale morphology of the ER surveyed by STED microscopy," *J. Cell Biol.* **218**(1), 83–96 (2019).
22. M. G. M. Velasco, M. Zhang, J. Antonello, P. Yuan, E. S. Allgeyer, D. May, O. M'Saad, P. Kidd, A. E. S. Barentine, V. Greco, J. Grutzendler, M. J. Booth, and J. Bewersdorf, "3D super-resolution deep-tissue imaging in living mice," *Optica* **8**(4), 442 (2021).
23. M. G. Giacomelli, "Optimal real-time resonant scanner linearization using filtered Hermite interpolation: code," Github, 2023, <https://github.com/mgiacomelli/dewarp>.